

Atty. Docket No. MS160276.1

METHOD AND SOFTWARE TOOLS FOR  
INTELLIGENT OPERATING SYSTEM  
INSTALLATION

by

Erin M. Bourke-Dunphy and Paul R. Fitzgerald

I hereby certify that the attached patent application (along with any other paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date November 29, 2000, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number EL782426180US addressed to the: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Himanshu S. Amin

(Typed or Printed Name of Person Mailing Paper)



(Signature of Person Mailing Paper)

**Title:           Method and Software Tool for Intelligent Operating System  
                  Installation**

**Technical Field**

5           The present invention relates generally to computer systems, and more particularly to methods and software tools for intelligent software component setup in computer systems.

**Background**

10           Software applications, operating systems, and/or services are commonly offered to consumers in the form of grouped packages or suites, in which two or more such software components are provided. Such suites are generally provided at a lower cost than the combined cost of the individual applications included therein. In addition, the software suites typically comprise an assortment of related and/or interoperable  
15 applications, operating systems, and/or services targeted to satisfy the software needs of a particular type of customer, such as small businesses. Different suites may thus be provided to satisfy various market needs, where certain segments of software consumers commonly require more than one application, operating system, or service. The suite may be provided in the form of a CDROM having the software component files (*e.g.*,  
20 application files and/or operating system files) and a setup or installation tool, which the user employs to setup the suite on the user's computer system.

          In addition to initial installation, the applications included within any particular suite package, as well as the software operating system (OS) residing in a customer's computer system, may be revised or improved from time to time, in order to provide new  
25 and/or improved features, and/or to provide new application components to an installed suite. Consumers typically are provided with such software revisions through upgrade packages. For instance, an upgrade may replace an existing software application with a newer version or revision. This may be sold to the customer in the form of a CDROM having the new application files and a setup or installation tool, which the user employs

to remove the old revision and install the new one. Upgrades are usually sold for major revisions of a software application or service.

The software suite offerings may be employed in single machines, as well as in networks of computers. Some such software suites are created specifically for small business settings, in which the target purchaser has only a small number of computers networked together (*e.g.*, less than 50). Other suites are designed to accommodate much larger network organizations, and thus provide a range of features beyond those provided in small business type software suites. For example, a small business suite may include a fixed, maximum number of client access licenses (CALs) at a fixed cost, whereas large business software suites may offer the user options to customize their license purchases.

While various software suites may be designed for different target consumer groups with differing needs, certain components (*e.g.*, operating system and/or application components) may be common to two or more such suites. This allows the software vendors to use the same components in a variety of suites, providing economic as well as logistical advantages. For instance, an operating system component may be adapted to operate in small business software suites as well as those designed for large businesses, thus reducing the vendor costs associated with developing and supporting two different operating systems.

Because some suites cost more than others, unscrupulous users may be inclined to purchase lower cost software suites (*e.g.*, those targeted toward small business applications) and to install one or more components thereof in system configurations for which they were not intended (*e.g.*, in a large organization), rather than purchasing the higher cost suite package, and the licenses associated therewith. Thus, there is a need to limit the functionality of software components in lower cost suites to prevent such unintended usage and to allow proper enforcement of software licenses, particularly where the lower cost and higher cost suites include common components.

Moreover, where a common component is used in two different suites, there may be one or more setup parameters for which a user must make a decision in one such suite, whereas no decision is needed for the same component in a second suite. Thus, there is a need for setup and/or installation tools, which provide for logical, high quality setup

prompting where a common component is included in two suites having different cost targets and/or functional capabilities.

### Summary

5           The following presents a simplified summary of the invention in order to provide a basic understanding of some aspects of the invention. This summary is not an extensive overview of the invention. It is intended to neither identify key or critical elements of the invention nor delineate the scope of the invention. Its sole purpose is to present some concepts of the invention in a simplified form as a prelude to the more detailed  
10           description that is presented later. The invention comprises a software tool and methodology, which address the above-mentioned problems associated with performing setup operations with respect to software components in a computer system. For example, the invention provides tools and methods which may be employed to prevent unscrupulous persons from employing one or more components from a low cost software  
15           suite in systems for which they were not intended, as well as to provide for proper licensing and logical setup experiences for legitimate users.

          A software setup tool and methodology are provided for performing a setup operation in a computer system. The setup operation may comprise, for example, installing an operating system, application, and/or other software component onto the  
20           system. The software tool queries or senses the value of a registry key or other system attribute and selectively modifies or customizes the installation according to the sensed attribute. For example, a registry key may be sensed in the computer system registry (e.g., via a direct query or by using an application program interface (API)). Based on the value of the registry key, one or more setup screens or prompts may be selectively  
25           rendered or suppressed from being rendered to the user via an interface. Alternatively or in combination, one or more additional prompts may be provided to the user according to the attribute. In addition, the software tool may selectively prompt a user for a setup related choice, or use an unprompted default setup parameter associated with one or more setup options according to the sensed attribute.

For instance, where a setup screen is adapted to prompt the user for a first or second choice, and the registry key value indicates that the second choice is invalid for the computer system, the setup screen may be unnecessary. In this case, the software tool may selectively use the first choice in installing or setting up the operating system, and  
5 may further selectively suppress the rendering of the setup screen to the user. In this fashion, the user is presented with a logical setup prompt flow tailored to the specific system in which the setup is occurring. Thus, where an operating system or other program is being installed in a system having an application suite which renders one or more of the operating system setup choices inapplicable or otherwise unnecessary, the  
10 software tool may easily sense an attribute, such as a registry key, indicating the installed application suite, and customize the operating system installation prompting accordingly.

In addition to selective prompting, the invention provides for tailoring of the setup according to the sensed attribute. For example, where a component (*e.g.*, an operating system) is common to both a low cost suite package and a higher cost suite package, the  
15 attribute may be used to selectively limit the functionality of the component in the lower cost suite. This may be advantageous to prevent unscrupulous purchasers of the lower cost suite from installing the component in a system other than that for which the suite was designed or intended. The invention may thus be employed in order to properly enforce licensing, for example, by limiting the number of client connections in a small  
20 business suite setup of an operating system component.

In accordance with one aspect of the invention, there is provided a software tool for installing one or more software components on a computer system, comprising a user interface component and a setup component. The user interface component is adapted to render one or more setup prompts to a user and to obtain a user command. For example,  
25 the user may be prompted to provide setup selections or other setup parameters. The setup component is adapted to selectively install the software component or components on the computer system according to the setup parameters, to sense an attribute associated with the computer system and to selectively provide setup prompting to the user interface component according to the attribute. The sensed attribute may comprise

any attribute associated with the target computer system, for example, a key or entry in the system registry or other data store.

The setup prompting may thus be customized for the particular computer system according to the sensed attribute, whereby one or more setup prompts may be suppressed from the user interface component, and/or one or more additional prompts may be provided. For example, where a setup prompt comprises a selection between two or more setup parameters, and only one of the possible selections is applicable to the system (as indicated by the attribute), the setup component may select the applicable setup parameter according to the attribute, and bypass rendering the setup prompt to the user.

One or more software components (*e.g.*, applications and/or operating systems) may then be installed and/or upgraded using the selected setup parameter without prompting the user to make unnecessary and/or confusing selections. In this way, one or more software components may be selectively installed and/or configured according to the attribute, which may indicate, for example, that the user has only purchased a low cost software suite, or otherwise has purchased only a limited number of licenses.

According to another aspect of the invention, there is provided a software setup tool for performing a setup operation in a computer system, comprising a setup component adapted to sense an attribute associated with the computer system (*e.g.*, a registry key) and to selectively provide one of standard prompting and special prompting to a user according to the attribute. The setup tool further comprises a user interface component adapted to render a plurality of setup prompts to a user and to obtain one or more setup parameters. The tool may also perform one or more setup operations according to the setup parameters and the attribute.

The setup component may selectively suppress at least one setup prompt from the user interface component if the attribute indicates the prompt is unnecessary. In addition, where an unnecessary setup prompt includes a selection between at least two setup parameters, the setup component may advantageously select one of the setup parameters according to the attribute. The tool may also provide one or more additional setup prompts to the user according to the attribute. Moreover, the setup component may advantageously tailor the setup of one or ore software components (*e.g.*, installation,

upgrade, and the like) according to the attribute. Thus, the invention prevents unauthorized or unintended use of a low cost software suite component, and provides for enforcement of licenses.

According to yet another aspect of the invention, there is provided a method of  
5 performing a setup operation in a computer system, which comprises sensing an attribute associated with the computer system (*e.g.*, a registry key), and selectively providing standard or special prompting to a user in association with the setup operation according to the attribute. The prompting may include rendering a plurality of setup prompts to a user, and obtaining one or more setup parameters, for example, via a user interface,  
10 whereafter one or more setup operations may be selectively performed according to the setup parameters.

Special prompting may be provided, for example, where the tool senses that a particular suite of software components is installed on the system. This customized or special prompting may comprise suppressing one or more one setup prompts from the  
15 user interface component (*e.g.*, where the attribute indicates the prompt is unnecessary) and/or rendering one or more additional setup prompts to the user interface. Special prompting in this regard, may further comprise rendering one or more alternative or substitute prompts to the user in lieu of one or more standard prompts, according to the attribute.

Another aspect of the invention provides a computer-readable medium  
20 comprising computer-executable instructions for sensing an attribute associated with the computer system, and selectively providing one of standard prompting and special prompting to a user in association with the setup operation according to the attribute. Still another aspect of the invention provides a data packet adapted to be transmitted  
25 between at least two processes, comprising a setup component adapted to sense an attribute associated with the computer system and to selectively provide one of standard prompting and special prompting to a user according to the attribute, and a user interface component adapted to render a plurality of setup prompts to a user and to obtain at least one setup parameter.

To the accomplishment of the foregoing and related ends, certain illustrative aspects of the invention are described herein in connection with the following description and the annexed drawings. These aspects are indicative, however, of but a few of the various ways in which the principles of the invention may be employed and the present invention is intended to include all such aspects and their equivalents. Other advantages and novel features of the invention may become apparent from the following detailed description of the invention when considered in conjunction with the drawings.

### **Brief Description Of The Drawings**

Fig. 1 is a flow diagram illustrating an exemplary method of performing a setup operation in a computer system in accordance with an aspect of the present invention;

Fig. 2 is a front elevation view of an exemplary user interface setup prompt which may be selectively suppressed in accordance with the invention;

Fig. 3 is a front elevation view of another exemplary user interface setup prompt which may be selectively suppressed in accordance with the invention;

Fig. 4 is a flow diagram illustrating another exemplary method of performing a setup operation in accordance with the invention;

Fig. 5 is a flow diagram illustrating another exemplary method of performing a setup operation in accordance with the invention;

Fig. 6 is a schematic diagram illustrating an exemplary software setup tool according to another aspect of the invention;

Fig. 7 is a flow diagram illustrating another exemplary method of performing a setup operation in a computer system according to the invention;

Fig. 8 is a flow diagram further illustrating the method of Fig. 7; and

Fig. 9 is a schematic block diagram illustrating an exemplary operating environment in which one or more aspects of the invention may be implemented.

### **Detailed Description**

The present invention is now described with reference to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following



description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It may be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. Moreover, well-known structures and devices are illustrated in some instances in block diagram form in order to facilitate description of the present invention.

Referring now to the drawings, the present invention comprises a software tool and methodology whereby the functionality of a software component may be selectively limited in lower cost suites to prevent unintended usage and to allow proper enforcement of software licenses, particularly where the lower cost and higher cost suites include common components. The invention further provides for logical, high quality setup prompting where a common component is included in two suites having different cost targets and/or functional capabilities.

In Fig. 1, an exemplary method 2 is illustrated for performing one or more setup operations in a computer system. Although the method of Fig. 1 is illustrated and described as a series of steps, it is to be understood and appreciated that the present invention is not limited by the order of steps, as some steps may, in accordance with the present invention, occur in different orders and/or concurrently with other steps from that shown and described herein. Moreover, not all illustrated steps may be required to implement a methodology in accordance with one or more aspects of the present invention.

The method 2 comprises sensing an attribute associated with the computer system, and selectively providing one of standard prompting and special prompting to a user in association with the setup operation according to an attribute, such as a registry key. Method 2 further selectively performs at least one setup operation according to one or more setup parameters and the attribute. In this way, one or more software components may be selectively installed and/or configured according to the attribute, which may indicate, for example, that the user has only purchased a low cost software suite, or otherwise has purchased only a limited number of licenses.

The method 2 may be employed, for example, by a software setup tool for installing and/or upgrading an operating system or other type software component. Beginning at step 4, an attribute (*e.g.*, a key or other entry in a computer system data store such as a registry) is sensed at step 6. For example, where the attribute is a registry key (not shown), the registry may be queried directly by the setup software tool, or the value of the key may be obtained via an API (not shown).

At step 8, a determination is made as to whether the attribute indicates that special prompting is appropriate. For example, the attribute may indicate that a software suite designed for small businesses is installed in the system. In this situation, there may be limitations placed on the functionality or licensing of one or more components of the installed suite (*e.g.*, applications or operating system). Where the component currently being installed or upgraded by the setup operation is common to other (*e.g.*, more costly) suites, the standard setup prompting for the component may include questions or prompts relating to functions not available in the installed suite. The invention prevents a user from accessing these functions by providing special or selective prompting according to the sensed attribute, as described in greater detail hereinafter.

If special prompting is not required at step 8, the standard prompting (*e.g.*, the normal prompting associated with the setup operation) is rendered to a user interface at step 10. Otherwise, the invention renders special prompting to the interface at step 12. As illustrated and described in greater detail hereinafter, special prompting may comprise suppressing one or more prompts from the user, rendering one or more additional prompts to the user, and/or selectively substituting one or more special prompts in place of or in lieu of one or more standard prompts. In this fashion, the setup experience is tailored to the system in which the setup is taking place by the advantageous sensing of the attribute associated therewith.

Thereafter, one or more setup parameters are obtained from the user interface at step 14, which are then used in performing one or more setup operations (*e.g.*, installation and/or upgrading of one or more software components) at step 16, after which the method 2 ends at step 18. The setup operations at step 16 may further be performed according to the attribute sensed at step 6. For example, one or more setup parameters may be

determined solely according to the attribute, such as where the installation of a small business oriented software suite indicates that an operating system must be installed in per server licensing mode, or that the operating system component must be installed in the root directory as the root domain controller.

5           Thus, where prompting of these setup parameters or selections (*e.g.*, per server licensing mode vs. per seat licensing mode, installation directory, and the like) is unnecessary, the setup tool may advantageously use appropriate default settings determined according to the attribute. It will be noted that this methodology 2 prevents the user from obtaining functionality or licenses where these have not been properly paid  
10 for, as well as provides for logical prompting without unnecessary prompts to choose from two or more selections where only one selection is possible or appropriate.

In order to further illustrate the concept of selective prompting, Figs. 2 and 3 illustrate two exemplary user prompts which may be selectively suppressed in accordance with an aspect of the present invention. The prompt 20 of Fig. 2 includes a licensing  
15 modes setup screen, which allows a user to select one of 'per server' licensing mode and 'per seat' licensing mode. The licensing mode selection via the prompt 20 may be employed as part of an operating system setup operation. However, depending on an attribute associated with the system in which the setup is taking place, this licensing mode selection may be inapplicable or unnecessary.

20           For example, a software suite (*e.g.*, targeted toward small business organizations) may be installed on the system which permits only 'per server' licensing mode. An attribute (*e.g.*, a registry key) associated with the system may indicate the existence of such a suite. Where a user knows of this requirement, the rendering of the prompt 20 may lead to user confusion, or an attempt to select 'per seat' licensing mode.

25           Accordingly, the invention provides for suppressing the prompt screen 20 during setup, and the use of 'per server' licensing mode as a default setup parameter. In addition, the setup may provide an indication to the user that 'per server' licensing mode is being employed, for example, by rendering a substitute prompt (not shown) to the user.

Another example is illustrated in Fig. 3, where a setup prompt 22 provides a  
30 selection of whether or not to configure the target computer system as a member in a

domain. This prompt may be selectively employed, for example, as part of an operating system setup operation. In a situation where an installed software suite (*e.g.*, such as a suite targeted for small business organizations) requires the system to be configured as a root domain controller, the invention may provide for suppressing the rendering of the prompt 22, and defaulting the setup operation to configure the system as the root domain controller in a workgroup entitled 'workgroup'. In this situation, the existence of the installed software suite may be determined by consulting a system attribute, such as a registry key. Thus, the user is not forced to ponder over potentially ambiguous information, and the setup experience is tailored to the system requirements.

As a further illustration of various aspects of the invention, another exemplary method 30 is illustrated in Fig. 4 for performing setup operations in a computer system. Although the method 30 is illustrated in Fig. 4 and described herein as a series of steps, the present invention is not limited by the illustrated ordering of steps. Thus, in accordance with the invention, some steps may occur in different orders and/or concurrently with other steps from that shown and described herein. Moreover, not all illustrated steps may be required to implement a methodology in accordance with the invention.

Beginning at step 32, a registry key is sensed at step 34, after which a prompt counter is set to 1 at step 36. Thereafter at decision step 38, a determination is made as to whether the sensed registry key indicates special prompting is required or otherwise appropriate. If not, the current (*e.g.*, standard) prompt associated with the setup operation is rendered to a user interface at step 40 and one or more selected setup parameters are obtained from the user interface at step 42. If special prompting is indicated by the attribute at step 38, one or more default parameters may be selected by the setup tool (*e.g.*, where the attribute indicates that no user selections are appropriate for the installed suite) at step 44. In this case, the tool may forgo or otherwise suppress rendering a prompt to the user.

It will be noted that alternative to, or in combination with the selective suppression of prompting, the invention further comprises rendering one or more additional prompts as appropriate according to the attribute. In addition, the invention

contemplates selectively substituting one or more special prompts for one or more standard prompts associated with a setup operation. For example, a standard prompt may allow a user to select one of three possible values for a setup parameter. Where the attribute renders only two of these three values applicable or otherwise appropriate to the system, a substitute prompt may be rendered to the user (*e.g.*, in lieu of the standard prompt), which comprises only the two applicable setup parameter values.

In this fashion, the user is not forced to ponder over a known inappropriate choice, and/or is spared the need to be notified (*e.g.*, via a further error prompt) that an erroneous selection has been made and to have to re-enter another selection. Thus, the invention provides for logical, consistent, and intelligent user prompting in association with the setup of a software component that may be common to various situations (*e.g.*, part of different software suites). Moreover, the invention may be employed to prevent the user from enabling certain functionality associated with a component being setup, where the user has not properly paid for such functionality (*e.g.*, the attribute indicates that only a fixed number of clients may be supported). It will be further recognized that the invention is applicable in association with many forms of user prompts, including but not limited to visual prompt screens, audio prompts, and the like.

Once standard prompting has been performed via steps 40 and 42 or one or more default setup parameters are selected (*e.g.*, in lieu of such prompting) at step 44, the method 30 proceeds to step 46 where the prompt counter is incremented. The prompt counter is then compared with a maximum (*e.g.*, determined according to the number of standard prompts associated with a setup operation) at decision step 48. If the maximum has not yet been exceeded at step 48, the method 30 returns to step 38 for processing of further prompting as described above. Once the maximum number of prompts have been thus processed, the setup is performed at step 50 according to the setup parameters (*e.g.*, one or more of which may have been determined or otherwise selected according to the attribute or registry key), and the method 30 ends at step 52.

In accordance with another aspect of the invention, Fig. 5 illustrates another exemplary method 60 for performing a setup operation in a computer system, which begins at step 62. Although the method 60 is illustrated in Fig. 5 and described herein as

a series of steps, the present invention is not limited by the illustrated ordering of steps. Thus, in accordance with the invention, some steps of the method 60 may occur in different orders and/or concurrently with other steps from that shown and described herein. Moreover, not all illustrated steps may be required to implement a methodology  
5 in accordance with the invention.

A registry key is sensed (*e.g.*, via direct query or API) at step 64, and a prompt counter is set initialized to 1 at step 66. At step 70, the current prompt is rendered to a user interface, after which one or more user selected parameters or parameter values are obtained from the user interface at step 72. At decision step 74, a determination is made  
10 as to whether or not the registry key indicates that special prompting is required or appropriate. If not, the prompt counter is incremented at step 76 and compared with a prompt maximum value at step 78.

If, however, the registry key indicates special prompting at step 74, the method 60 proceeds to step 80 where one or more additional prompts may be rendered to the user  
15 interface, and one or more additional setup parameters may be obtained therefrom at step 82. The invention thus provides for selectively suppressing at least one setup prompt from the user interface component if the attribute indicates the at least one setup prompt is unnecessary, and/or rendering at least one additional setup prompt to the user interface according to the attribute (*e.g.*, the sensed registry key).

The method 60 proceeds in this fashion until the maximum prompt value has been exceeded at step 78, whereafter one or more setup operations (*e.g.*, installing and/or  
20 upgrading one or more software components in a computer system) are performed at step 84. The setup may thus be performed at step 84 according to the setup parameters (*e.g.*, one or more of which may have been determined or otherwise selected according to the  
25 attribute or registry key), after which the method 60 ends at step 86.

Although the methods 30 and 60 of Figs. 4 and 5, respectively, have been illustrated as providing for selectively suppressing at least one setup prompt from the user interface component and/or rendering at least one additional setup prompt to the user interface according to the attribute (*e.g.*, the sensed registry key), the invention is not  
30 limited to the illustrated methodologies. For example, the invention may further

comprise (*e.g.*, alternatively or in combination therewith) rendering at least one substitute setup prompt to the user interface (*e.g.*, in lieu of one or more standard prompts) according to the attribute.

Referring now to Fig. 6, an exemplary software tool 100 is illustrated for performing a setup operation. The setup operation may comprise, for example, installing and/or upgrading one or more software components on a computer system 102. The software components may include any type thereof, such as an operating system 104, and/or one or more applications such as server application number 1 106, server application number 2 108, and server application number N 110, where N is an integer. The software components may, but need not be, part of the tool 100. The setup tool 100 comprises a user interface component 112 adapted to render a plurality of setup prompts (not shown) to a user 114 and to obtain one or more user commands or selections therefrom.

In addition, the tool 100 comprises a setup component 116 adapted to selectively install and/or upgrade one or more software components (*e.g.*, components 104, 106, 108, and/or 110) on the computer system 102 according to the user command. Setup component 116 may further be adapted to sense an attribute such as one or more registry keys 120, 122, and/or 124 associated with the computer system 102. For example, the registry keys 120, 122, and 124 may be included in a data store such as a registry 130. In this regard, the setup component 116 of the setup tool 100 may sense or otherwise obtain the registry key (*e.g.*, key 120) or a value associated therewith via direct query to the registry 130 or via an API 132.

It will be further noted that one or more of the registry keys 120, 122, and/or 124 may include information related to software components 140 installed on the system 102. For example, the registry key 120 may indicate that a software suite has been previously installed on the computer system 102, which is designed for use by small businesses. The setup tool 100 thus may advantageously provide targeted prompting according to the key 120, whereby standard prompts associated with a software component being installed or upgraded (*e.g.*, components 104, 106, 108 and/or 110) may be selectively suppressed,

additional prompts may be provided, and/or substitute prompts may be rendered in lieu of one or more standard prompts, in accordance with the invention.

This avoids unnecessary prompting, and provides for limiting the functionality associated with one or more of the installed or upgraded components. For example, the  
5 tool 100 may use default values in lieu of obtaining elections from the user 114 where the registry key 120 indicates that no selection is needed or appropriate for a particular setup parameter.

Figs. 7 and 8 illustrate another exemplary method 200 of performing a setup operation (*e.g.*, installing or upgrading a software component) in a computer system (*e.g.*,  
10 system 102 of Fig. 6). Although the method 200 is illustrated in Figs. 7 and 8 and described herein as a series of steps, the present invention is not limited by the illustrated ordering of steps. Thus, in accordance with the invention, some steps may occur in different orders and/or concurrently with other steps from that shown and described herein. Moreover, not all illustrated steps may be required to implement a methodology  
15 in accordance with the invention. Beginning at step 202, a registry key (or other attribute associated with the computer system) is sensed at step 204 (*e.g.*, via direct querying of a registry or other data store or via an API). Thereafter, one or more standard prompts may be rendered to a user (*e.g.*, via a user interface) at steps 206, 208, and 210.

At decision step 212, a determination is made as to whether the registry key  
20 sensed at step 204 indicates or otherwise calls for special prompting to be employed in the setup operation. For example, the registry key (*e.g.*, key 120 of Fig. 6) may indicate that a small business suite is installed in the system (*e.g.*, components 140 of system 102). This suite may require that an operating system component being setup (*e.g.*, installed and/or upgraded) should be installed in a per server licensing mode, with a fixed  
25 number of client access licenses (CALs) of ten. Proceeding with this example, if no special prompting is indicated by the sensed key, the method 200 proceeds to step 214, where a licensing prompt is rendered to the user (*e.g.*, to ascertain how many licenses the user desires, and whether the licensing should be done on a per seat basis or on a per server basis).



However, if the small business suite is installed in the system, or the sensed key otherwise indicates that special prompting is appropriate, the method 200 proceeds to step 216, whereby the licensing prompt is bypassed or suppressed, and default values are used for the setup parameters associated with the licensing mode and the number of licenses. Thus, in this example, the setup tool may default these parameters to installation of the operating system in per server license mode, with 10 licenses.

Thereafter, one or more further standard prompts may be rendered to the user at steps 218, 220, and 222. At decision step 224, a determination is made as to whether special prompting is again required or appropriate. For example, the key sensed at step 204 may indicate that the small business software suite installed on the target system requires that the operating system be installed as the root domain controller (*e.g.*, where the small business suite is not designed to support multiple domains or workgroups). In this case, rather than allowing the user to incorrectly or erroneously select to install (or upgrade) the operating system software component as other than the root domain controller, the method 200 may selectively set a default setup parameter such that the OS is installed as part of a workgroup named "Workgroup" to ensure that the installed or upgraded OS is the root domain controller, and may further bypass or suppress rendering a domain/workgroup prompt which would have otherwise been performed at step 228.

Thereafter, the method 200 proceeds through steps 230 and 232, where further (standard) user prompting is performed. The operating system may then be installed at step 234 using prompted parameters and default setup parameters, after which the method 200 ends at step 236.

In order to provide a context for the various aspects of the invention, Fig. 9 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the various aspects of the present invention may be implemented. While the invention has been described above in the general context of software tools and computer-executable instructions of a computer program that runs on a computer and/or computers, those skilled in the art will recognize that the invention also may be implemented in combination with other program modules. Generally, program

modules include routines, programs, components, data structures, *etc.* that perform particular tasks and/or implement particular abstract data types.

Moreover, those skilled in the art will appreciate that the inventive methods may be practiced with other computer system configurations, including single-processor or  
5 multiprocessor computer systems, minicomputers, mainframe computers, as well as personal computers, hand-held computing devices, microprocessor-based or programmable consumer electronics, and the like. The illustrated aspects of the invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

10 However, some, if not all aspects of the invention can be practiced on stand-alone computers. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to Fig. 9, an exemplary environment 310 for implementing various aspects of the invention includes a computer 312, including a processing unit 314, a  
15 system memory 316, and a system bus 318 that couples various system components including the system memory to the processing unit 314. The processing unit 314 may be any of various commercially available processors, including but not limited to Intel x86, Pentium®, and compatible microprocessors from Intel and others, including Cyrix, AMD and Nexgen; Alpha®, from Digital; MIPS®, from MIPS Technology, NEC, IDT,  
20 Siemens, and others; and the PowerPC®, from IBM and Motorola. Dual microprocessors and other multi-processor architectures also can be used as the processing unit 314.

The system bus 318 may be any of several types of bus structure including a memory bus or memory controller, a peripheral bus, and a local bus using any of a  
25 variety of conventional bus architectures such as PCI, VESA, Microchannel, ISA, and EISA, to name a few. The computer 312 memory may include read only memory (ROM) 320 and random access memory (RAM) 322. A basic input/output system (BIOS), containing the basic routines that help to transfer information between elements within the computer 312, such as during start-up, is stored in ROM 320.

The computer 312 further includes a hard disk drive 324, a magnetic disk drive 326, *e.g.*, to read from or write to a removable disk 328, and an optical disk drive 330, *e.g.*, for reading a CD-ROM disk 332 or to read from or write to other optical media. The hard disk drive 324, magnetic disk drive 326, and optical disk drive 330 are connected to the system bus 318 by a hard disk drive interface 334, a magnetic disk drive interface 336, and an optical drive interface 338, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions, etc. for the computer 312, including for the storage of broadcast programming in a suitable digital format.

Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD-ROM, it will be appreciated by those skilled in the art that other types of media which are readable by a computer, such as zip drives, magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, may also be used in the exemplary operating environment, and further that any such media may contain computer-executable instructions for performing the methods of the present invention.

A number of program modules may be stored in the drives and RAM 322, including an operating system 340, one or more application programs 342, other program modules 344, and program data 346. The operating system 340 in the illustrated computer is, for example, the "Microsoft Windows NT" operating system, although it is to be appreciated that the present invention may be implemented with other operating systems or combinations of operating systems, such as UNIX, LINUX, etc.

A user may enter commands and information into the computer 312 through a keyboard 348 and a pointing device, such as a mouse 350. Other input devices (not shown) may include a microphone, an IR remote control, a joystick, a game pad, a satellite dish, a scanner, or the like. These and other input devices are often connected to the processing unit 314 through a serial port interface 352 that is coupled to the system bus 318, but may be connected by other interfaces, such as a parallel port, a game port, a universal serial bus ("USB"), an IR interface, etc. A monitor 354 or other type of display device is also connected to the system bus 318 *via* an interface, such as a video adapter

356. In addition to the monitor, a computer typically includes other peripheral output devices (not shown), such as speakers, printers etc.

The computer 312 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer(s) 358. The remote computer(s) 358 may be a workstation, a server computer, a router, a personal computer, microprocessor based entertainment appliance (*e.g.*, a WebTV client system), a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 312, although, for purposes of brevity, only a memory storage device 360 is illustrated. The logical connections depicted include a local area network (LAN) 362 and a wide area network (WAN) 364. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 312 is connected to the local network 362 through a network interface or adapter 366. When used in a WAN networking environment, the computer 312 typically includes a modem 368, or is connected to a communications server on the LAN, or has other means for establishing communications over the WAN 364, such as the Internet. The modem 368, which may be internal or external, is connected to the system bus 318 *via* the serial port interface 352 (*e.g.*, for communications over POTS). The modem 368, alternatively, may be connected to the system bus 318 *via* the network interface or adapter 366 (*e.g.*, for communication over DSL, cable, satellite, etc.). In a networked environment, program modules depicted relative to the computer 312, or portions thereof, may be stored in the remote memory storage device 360. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Although the invention has been shown and described with respect to certain implementations, it will be appreciated that equivalent alterations and modifications will occur to others skilled in the art upon the reading and understanding of this specification and the annexed drawings. In particular regard to the various functions performed by the above described components (assemblies, devices, circuits, systems, etc.), the terms

(including a reference to a "means") used to describe such components are intended to correspond, unless otherwise indicated, to any component which performs the specified function of the described component (*i.e.*, that is functionally equivalent), even though not structurally equivalent to the disclosed structure, which performs the function in the  
5 herein illustrated exemplary applications and implementations of the invention.

In addition, while a particular feature of the invention may have been disclosed with respect to only one of several aspects or implementations of the invention, such a feature may be combined with one or more other features of the other implementations as may be desired and advantageous for any given or particular application. Furthermore, to  
10 the extent that the terms "includes", "including", "has", "having", and variants thereof are used in either the detailed description or the claims, these terms are intended to be inclusive in a manner similar to the term "comprising" and its variants.

What has been described above includes examples of the present invention. It is, of course, not possible to describe every conceivable combination of components or  
15 methodologies for purposes of describing the present invention, but one of ordinary skill in the art may recognize that many further combinations and permutations of the present invention are possible. Accordingly, the present invention is intended to embrace all such alterations, modifications and variations that fall within the spirit and scope of the appended claims.